



Subscribe Register Login
(Full Service) (Limited Service, Free)

Search: ☒ The ACM Digital Library ☐ The Guide

branch prediction <and> profile <and> probability grouping

THE ACM DIGITAL LIBRARY

Feedback

Terms used branch prediction and profile and probability grouping

Sort results
by

relevance

☒ Save results to a Binder

Try

Search Tips

Try

☐ Open results in a new window

Display results

expanded form

Results 1 - 20 of 200

Result page: **1** 2 3 4 5 6 7 8 9

Best 200 shown

1 The Accuracy of Initial Prediction in Two-Phase Dynamic Binary Translator

Youfeng Wu, Mauricio Breternitz, Justin Quek, Orna Etzion, Jesse Fang

March 2004 Proceedings of the international symposium on Code generation and runtime optimization

Full text available: pdf(234.04 KB)

Additional Information: full citation

Dynamic binary translators use a two-phase approach to identify and optimize. In the first step (profiling phase), blocks of code are interpreted or quickly translated to obtain information for the blocks. In the second phase (optimization phase), frequent regions and advanced optimizations are applied on them. This approach implicitly assumes that a block is representative of the block it ...

2 Embedded systems: applications, solutions and techniques (EMBS): Assessing coincident failures and usage-profiles on the reliability of embedded control

Frederick T. Sheldon, Kshamta Jerath

March 2004 Proceedings of the 2004 ACM symposium on Applied computing

Full text available: pdf(327.91 KB)

Additional Information: full citation, abstract, re


The increasingly ubiquitous use of embedded systems to manage and control complex lives makes us more vulnerable than ever before. Knowing how reliable systems are necessary especially for safety, mission and infrastructure critical applications. A compositional modeling method for assessing reliability based on characteristic failure patterns illustrate this using a classic embedded control sys ...

Keywords: design, measurement, performance, reliability

3 Static branch frequency and program profile analysis

Youfeng Wu, James R. Larus

November 1994 Proceedings of the 27th annual international symposium on M

Full text available:  pdf(1.26 MB)

Additional Information: full citation, abstract, references, cit

Program profiles identify frequently executed portions of a program, which are programmers and compilers the greatest benefit. Compilers, however, infrequently profiling a program requires a programmer to instrument and run the program compiler to statically estimate program profiles. This paper presents several n prediction and profiling. The first ...

4 From profiles to patterns and back again: a branch and bound algorithm for Eleazar Eskin

March 2004 Proceedings of the eighth annual international conference on Compu

Full text available:  pdf(197.74 KB)


Additional Information: full citation, abstract, referen

An important part of deciphering gene regulatory mechanisms is discovering t many cases, these sites can be detected because they are often overrepresented of the overrepresented signals in sequences, or *motif-finding* has become a ce There are two major computational frameworks for attacking the motif finding representation of the signals. The most ...

Keywords: motif-finding, patterns, profiles, transcription factor binding sites

5 Corpus-based static branch prediction

Brad Calder, Dirk Grunwald, Donald Lindsay, James Martin, Michael Mozer, Benj
June 1995 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1995 confere
and implementation, Volume 30 Issue 6

Full text available:  pdf(1.35 MB)

Additional Information: full citation, abstract, references, ci

Correctly predicting the direction that branches will take is increasingly impor architectures. The name program-based branch prediction is given to static br their prediction on a program's structure. In this paper, we investigate a new prediction that uses a body of existing programs to predict the branch behavior approach to program-based....

6 Evidence-based static branch prediction using machine learning

Brad Calder, Dirk Grunwald, Michael Jones, Donald Lindsay, James Martin, Michi
January 1997 ACM Transactions on Programming Languages and Systems (TOPL

Full text available:  pdf(515.50 KB)

Additional Information: full citation, abstract, references, c

Correctly predicting the direction that branches will take is increasingly impor
architectures. The name program-based branch prediction is given to static br
their prediction on a program's structure. In this article, we investigate a new
prediction that uses a body of existing programs to predict the branch behavio
approach to program-ba ...

Keywords: branch prediction, decision trees, machine learning, neural network
optimization

7 Array regrouping and structure splitting using whole-program reference affi

Yutao Zhong, Maksim Orlovich, Xipeng Shen, Chen Ding

June 2004 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2004 confere
and implementation, Volume 39 Issue 6

Full text available:  pdf(202.16 KB)

Additional Information: full citation, abstract, referen

While the memory of most machines is organized as a hierarchy, program dat
space. This paper defines a model of *reference affinity*, which measures how c
together in a reference trace. It proves that the model gives a hierarchical pai
set of all data with the weakest affinity. At the bottom is each data element w
theoretical model, the paper p ...

Keywords: array regrouping, program locality, program transformation, refere
splitting, volume distance

8 Probabilistic discovery of overlapping cellular processes and their regulatio

Alexis Battle, Eran Segal, Daphne Koller

March 2004 Proceedings of the eighth annual international conference on Compu

Full text available:  pdf(259.52 KB)

Additional Information: full citation, abstract, referenc


Many of the functions carried out by a living cell are regulated at the transcrip
expressed when they are needed. Thus, to understand biological processes, it
cell's transcriptional network. In this paper, we propose a novel probabilistic n
identifying overlapping biological processes and the regulatory mechanism coi
of our approach is that we a ...

Keywords: cellular processes, gene regulation, probabilistic relational models

9 Branch prediction for free

Thomas Ball, James R. Larus

June 1993 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1993 conference and implementation, Volume 28 Issue 6

Full text available:  pdf(1.49 MB)

Additional Information: full citation, abstract, references, ci

Many compilers rely on branch prediction to improve program performance by and by aiding in scheduling instructions. Profile-based predictors require a time compile-profile-compile cycle in order to make predictions. We present a program that performs well for a large and diverse set of programs written in C and Fortran. This paper presents an analysis to pre ...

10 Using branch handling hardware to support profile-driven optimization

Thomas M. Conte, Burzin A. Patel, J. Stan Cox

November 1994 Proceedings of the 27th annual international symposium on Microarchitecture

Full text available:  pdf(954.48 KB)



Additional Information: full citation, abstract, references, ci

Profile-based optimizations can be used for instruction scheduling, loop scheduling, and instruction cache performance enhancement. However, these techniques are not used by hardware vendors because programs instrumented for profiling run 2–30 times slower. A compile-run-recompile sequence is required, and a test input suite must be compiled. This paper proposes using existing branch handling hardware to support profile-driven optimization.

11 Technical papers: consistency management and quality assurance: Automated failure reports

Andy Podgurski, David Leon, Patrick Francis, Wes Masri, Melinda Minch, Jiayang

May 2003 Proceedings of the 25th International Conference on Software Engineering

Full text available:  pdf(1.06 MB)  Publisher Site

Additional Information: full citation, abstract, references, ci

This paper proposes automated support for classifying reported software failures and diagnosing their causes. A classification strategy is presented that involves unsupervised pattern classification and multivariate visualization. These techniques are used to group together failures with the same or similar causes to assess the frequency and severity of failures.

12 Interfaces economics and computer science: Costly valuation computation

Kate Larson, Tuomas Sandholm

July 2001 Proceedings of the 8th conference on Theoretical aspects of rationality and bargaining

Full text available:  pdf(862.73 KB)


Additional Information: full citation, abstract, references, ci

We investigate deliberation and bidding strategies of agents with costly computation who are participating in auctions. The agents do not compute valuations for the items being auctioned. Instead they devote computation to compute their valuations. We present a normative model of both deliberation and bidding actions of agents are incorporated into strategies for a standard auction protocols. We show that ...

13 Predicting conditional branch directions from previous runs of a program

Joseph A. Fisher, Stefan M. Freudenberger

September 1992 ACM SIGPLAN Notices , Proceedings of the fifth international conference on programming languages and operating systems, Volume 27 Issue 5

Full text available:  pdf(1.13 MB)

Additional Information: full citation, references, citations

14 Improving semi-static branch prediction by code replication

Andreas Krall

June 1994 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1994 conference on programming languages and implementation, Volume 29 Issue 6

Full text available:  pdf(752.92 KB)

Additional Information: full citation, abstract, references, citations

Speculative execution on superscalar processors demands substantially better branch prediction than was previously available. In this paper we present code replication techniques that improve branch prediction to a level comparable to dynamic branch prediction schemes. We provide information about the correlation between different branches and about the code replication outcomes of a single branch. Using this information we can improve branch prediction by replicating code blocks that are likely to be executed multiple times.

15 Active learning for automatic classification of software behavior

James F. Bowring, James M. Rehg, Mary Jean Harrold

July 2004 ACM SIGSOFT Software Engineering Notes , Proceedings of the 2004 ACM SIGSOFT Software testing and analysis, Volume 29 Issue 4

Full text available:  pdf(567.57 KB)

Additional Information: full citation, abstract, references, citations



A program's behavior is ultimately the collection of all its executions. This collection is generally unbounded. Thus it is especially suited to statistical analysis and machine learning. The focus of this paper is on the automatic classification of program behavior using machine learning. Since classifiers for software engineering adopts a classical *batch-learning* approach, we propose an *active-learning* paradigm for ...

Keywords: Markov models, machine learning, software behavior, software testing

16 Predicting data cache misses in non-numeric applications through correlation

Todd C. Mowry, Chi-Keung Luk

December 1997 Proceedings of the 30th annual ACM/IEEE international symposium on microarchitecture

Full text available:  pdf(876.36 KB)  Publisher Site

Additional Information: full citation, abstract, references, citations


To maximize the benefit and minimize the overhead of software-based latency prediction, we apply them precisely to the set of dynamic references that suffer cache misses. We provide information about the correlation between different branches and about the code replication outcomes of a single branch. Using this information we can improve branch prediction by replicating code blocks that are likely to be executed multiple times. To maximize the benefit and minimize the overhead of software-based latency prediction, we apply them precisely to the set of dynamic references that suffer cache misses. We provide information about the correlation between different branches and about the code replication outcomes of a single branch. Using this information we can improve branch prediction by replicating code blocks that are likely to be executed multiple times.

Keywords: profiling, cache miss prediction, correlation, non-numeric applications

17 Software reliability and dependability: a roadmap

Bev Littlewood, Lorenzo Strigini

May 2000 Proceedings of the Conference on The Future of Software Engineering

Full text available:  pdf(1.57 MB)

Additional Information: full citation, references, citations, index t

Keywords: COTS reliability, dependability modelling and assessment, diversity

18 Better exploration of region-level value locality with integrated computation

Youfeng Wu, Dong-Yuan Chen, Jesse Fang

May 2001 ACM SIGARCH Computer Architecture News , Proceedings of the 28th a
Computer architecture, Volume 29 Issue 2

Full text available:  pdf(940.95 KB)

Additional Information: full citation, abstract, references,

*Computation-reuse and value-prediction are two recent techniq
microprocessor performance by exploiting value localities. They
dependence limit in traditional processors. In this paper, we pr
multithreading scheme in which the same hardware can be effic
reuse and value prediction. For the SpecInt95 benchmarks, our
integrated approach significantly out-performs either c ...*

19 Research sessions: data mining applications: Cost-based labeling of group

Lei Chen, Zheng Huang, Raghu Ramakrishnan

June 2004 Proceedings of the 2004 ACM SIGMOD international conference on M

Full text available:  pdf(351.21 KB)



Additional Information: full citation, abstract,

We make two main contributions in this paper. First, we motivate and introdu
that arise in labeling a group of mass spectra, specifically for analysis of atmo
applications to market-basket datasets. This builds upon other recent work in
labeling a single spectrum, and is motivated by the advent of a new generatio
Spectrometers, which are capable of generating ma ...

20 Discourse segmentation by human and automated means

Rebecca J. Passonneau, Diane J. Litman

March 1997 Computational Linguistics, Volume 23 Issue 1




Full text available:  pdf(2.71 MB)  Publisher Site

Additional Information: full citation, a

The need to model the relation between discourse structure and linguistic feat
acknowledged in the literature on discourse. However, there is only weak cons
structure are, or the criteria for recognizing and generating them. We present
using a corpus of spontaneous, narrative monologues. The first part of our pag
validating multitutterance units ...

The ACM Portal is published by the Association for Computing Machinery. C

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Cont](#)

Useful downloads:  Adobe Acrobat  QuickTime  Windows Medi



Subscribe Register Login
(Full Service) (Limited Service, Free)

Search: ☒ The ACM Digital Library ☐ The Guide

branch prediction <and> switch statement <and> case statement

THE ACM DIGITAL LIBRARY

Feedback

Terms used branch prediction and switch statement and case statement

Sort results
by

relevance

☒ Save results to a Binder

Try

☒ Search Tips

Try

☐ Open results in a new window

Display results

expanded form

Results 1 - 20 of 200

Result page: **1** 2 3 4 5 6 7 8 9

Best 200 shown

1 Automatic data layout for distributed-memory machines

Ken Kennedy, Ulrich Kremer

July 1998 ACM Transactions on Programming Languages and Systems (TOPLAS)

Full text available: pdf(633.20 KB)

Additional Information: full citation, abstract, references, citir

The goal of languages like Fortran D or High Performance Fortran (HPF) is to provide a machine-independent parallel programming model. After the algorithm selection phase, the intellectual challenge in writing an efficient program in such languages. The problem is the target compilation system, the target machine, the problem size, and the choice of a good layout extremel ...

Keywords: high performance Fortran

2 Static correlated branch prediction

Cliff Young, Michael D. Smith

September 1999 ACM Transactions on Programming Languages and Systems (TOPLAS)

Full text available: pdf(508.49 KB)

Additional Information: full citation, abstract, references,


Recent work in history-based branch prediction uses novel hardware structure to increase branch prediction accuracy. Branch correlation occurs when the outcome of a branch is accurately predicted by observing the outcomes of previously executed branches. In this article, we show how to instrument a program so that it is practical to collect branch correlation data where branch correlation occurs ...

Keywords: branch correlation, branch prediction, path profiling, profile-driven

3 Target prediction for indirect jumps

Po-Yung Chang, Eric Hao, Yale N. Patt

May 1997 ACM SIGARCH Computer Architecture News , Proceedings of the 24th annual computer architecture conference, Volume 25 Issue 2

Full text available:  pdf(1.35 MB)


Additional Information: full citation, abstract, references, citations

As the issue rate and pipeline depth of high performance superscalar processors work issued also increases. Because speculative work must be thrown away in wide-issue, deeply pipelined processors must employ accurate branch prediction to achieve performance potential. Many existing branch prediction schemes are capable of handling conditional branches. However, these schemes are ...

4 A Survey of Some Theoretical Aspects of Multiprocessing

J. L. Baer

January 1973 ACM Computing Surveys (CSUR), Volume 5 Issue 1

Full text available:  pdf(4.05 MB) Additional Information: full citation, references, citations, index terms

5 Efficient and effective branch reordering using profile data

Minghui Yang, Gang-Ryung Uh, David B. Whalley

November 2002 ACM Transactions on Programming Languages and Systems (TOPLAS)

Full text available:  pdf(852.50 KB)

Additional Information: full citation, abstract, references, citations



The conditional branch has long been considered an expensive operation. The number of conditional branches executed often results in a substantial performance penalty. This paper presents a code-improving transformation to reorder sequences of conditional branches to reduce the number of conditional branches executed. The goal is to obtain an order of execution that minimizes the number of conditional branches executed. ...

Keywords: Conditional branches, branch reordering, profiling

6 Technical papers: testing I: Improving test suites via operational abstraction


Michael Harder, Jeff Mellen, Michael D. Ernst



May 2003 Proceedings of the 25th International Conference on Software Engineering (ICSE)


Full text available:  pdf(1.31 MB)  Publisher Site

Additional Information: full citation, abstract, references, citations

This paper presents the operational difference technique for generating, augmenting, and analyzing test cases. The technique is analogous to structural code coverage techniques, but it operates on the semantic domain of program properties rather than the syntactic domain of program text. The operational difference technique generates test cases; it assumes only the existence of a source of test cases. The technique generates test cases (which describe observed behavior) ...

- 7 A two-layer library-based approach to synthesis of analog systems from VHDL**
Alex Doboli, Nagu Dhanwada, Adrian Nunez-Aldana, Ranga Vemuri
April 2004 ACM Transactions on Design Automation of Electronic Systems (TODAES)
Full text available:  pdf(658.00 KB) Additional Information: full citation, abstract, references
This paper presents a synthesis methodology for analog systems described using VHDL. The methodology produces net-lists of analog components that are selected from a library, and the synthesis of the AC response, signal to noise ratio, dynamic range, area) are optimized. The methodology is bridged using a two-layered methodology. The first layer is component synthesis and constrain ...


Keywords: Analog synthesis, VHDL-AMS, branch-and-bound, genetic algorithm
- 8 Improving performance by branch reordering**
Minghui Yang, Gang-Ryung Uh, David B. Whalley
May 1998 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1998 conference on Programming Language Design and Implementation, Volume 33 Issue 5
Full text available:  pdf(1.51 MB) Additional Information: full citation, abstract, references, citations
The conditional branch has long been considered an expensive operation. The cost has increased as recently designed machines are now relying on deeper pipelines. The number of conditional branches executed can often result in a substantial performance penalty. A code-improving transformation to reorder sequences of conditional branches. Branches that are reordered are detected in the compiler ...
- 9 Compiler transformations for high-performance computing**
David F. Bacon, Susan L. Graham, Oliver J. Sharp
December 1994 ACM Computing Surveys (CSUR), Volume 26 Issue 4
Full text available:  pdf(6.32 MB) Additional Information: full citation, abstract, references, citations
In the last three decades a large number of compiler transformations for optimizing compilers have been implemented. Most optimizations for uniprocessors reduce the number of instructions. Transformations based on the analysis of scalar quantities and data-flow techniques. For high-performance superscalar, vector, and parallel processors maximize parallelism. Transformations that rely on tracking the properties of data ...

Keywords: compilation, dependence analysis, locality, multiprocessors, optimizing compilers, vectorization
- 10 Subrecursive Programming Languages, Part I: efficiency and program structure**
Robert L. Constable, Allan B. Borodin
July 1972 Journal of the ACM (JACM), Volume 19 Issue 3
Full text available:  pdf(1.89 MB) Additional Information: full citation, references, citations, index terms

11 Predicting indirect branches via data compression

John Kalamatianos, David R. Kaeli

November 1998 Proceedings of the 31st annual ACM/IEEE international symposium



Full text available:  pdf(1.24 MB)

Additional Information: full citation, references, citations, index te

12 Technical papers: software understanding: Tools for understanding the beh

André Marburger, Bernhard Westfechtel

May 2003 Proceedings of the 25th International Conference on Software Engi

Full text available:  pdf(2.23 MB)  Publisher Site

Additional Information: full cita

Many methods and tools for the reengineering of software systems have been domain-specific requirements of telecommunication systems have not been ac designed in a process- rather than in a data-centered way. Furthermore, analy is a key to system understanding. In this paper, we report on tools for the ree systems which we have developed in close cooperation wi ...

13 Selection conditions in main memory

Kenneth A. Ross

March 2004 ACM Transactions on Database Systems (TODS), Volume 29 Iss

Full text available:  pdf(296.54 KB)

Additional Information: full citation, abstract, referenc


We consider the fundamental operation of applying a compound filtering cond memories available cheaply, systems may choose to keep the data entirely in query and/or update performance. The design of a data-intensive algorithm in account the architectural characteristics of modern processors, just as a disk-l physical characteristics of disk devices. An importa ...

Keywords: Branch misprediction

14 Predicting conditional branch directions from previous runs of a program

Joseph A. Fisher, Stefan M. Freudenberger

September 1992 ACM SIGPLAN Notices , Proceedings of the fifth international conf programming languages and operating systems, Volume 27 Issu

Full text available:  pdf(1.13 MB)

Additional Information: full citation, references, citations

15 Static program analysis: Using redundancies to find errors

Yichen Xie, Dawson Engler

November 2002 Proceedings of the 10th ACM SIGSOFT symposium on Foundatio

Full text available:  pdf(265.85 KB)

Additional Information: full citation, abstract, referen

This paper explores the idea that redundant operations, like type errors, comr experimentally test this idea by writing and applying four redundancy checker many errors. We then use these errors to demonstrate that redundancies, eve with the presence of traditional hard errors (e.g., null pointer dereferences, ur how flagging redundant operations gives a ...

Keywords: error detection, extensible compilation

16 Parametric analysis for adaptive computation offloading

Cheng Wang, Zhiyuan Li

June 2004 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2004 confere and implementation, Volume 39 Issue 6

Full text available:  pdf(256.63 KB)

Additional Information: full citation, abstract, referen

Many programs can be invoked under different execution options, input param execution contexts may lead to strikingly different execution instances. The o to the execution instances. In this paper, we show how to use parametric prog the optimization problem of computation offloading. Computation offloading ha improve performance and energy saving ...

Keywords: adaptive optimization, computation offloading, distributed system, program partitioning, program profiling, program transformation

17 Comparative efficiency of general and residual parsers

Frank G. Pagan

April 1990

ACM SIGPLAN Notices, Volume 25 Issue 4

Full text available:  pdf(546.43 KB)

Additional Information: full citation, abstract,

Some fundamentals of the partial computation concept are concisely reviewed relationship between table-driven, general syntactic analyzers and source-lang analyzers. A manual methodology for converting general parsers into generat using an LL(1) parser as a detailed example. The results of several experimen space efficiencies of different general parsers wi ...

18 Compiling Esterel into sequential code

Stephen A. Edwards

June 2000

Proceedings of the 37th conference on Design automation

Full text available:  pdf(152.63 KB)

Additional Information: full citation, abstract, references,

Embedded real-time software systems often need fine-grained parallelism and typical real-time operating systems do not provide. The Esterel language has slow code for large programs. This paper presents the first Esterel compiler at programs. It can produce code half the size and up to a hundred times faster. Esterel's semantics allow th ...

19 Structured Programming with go to Statements

Donald E. Knuth

December 1974

ACM Computing Surveys (CSUR), Volume 6 Issue 4


Full text available:  pdf(3.02 MB) Additional Information: full citation, references, citings, index terms

20 Attention and integration: Learning and reasoning about interruption

Eric Horvitz, Johnson Apacible

November 2003

Proceedings of the 5th international conference on Multimodal

Full text available:  pdf(1.07 MB)

Additional Information: full citation, abstract, references,

We present methods for inferring the cost of interrupting users based on mult information generated by interactions with computing devices, visual and acoi online calendars. Following a review of prior work on techniques for deliberati associated with notifications, we introduce methods for learning models from expected cost of interruption for a user. We desc ...




Keywords: cognitive models, divided attention, interruption, notifications

Results 1 - 20 of 200

Result page: **1** 2 3 4 5 6 7

The ACM Portal is published by the Association for Computing Machinery. C

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Cont](#)

Useful downloads:  Adobe Acrobat  QuickTime  Windows Medi